```
OUT_M(i)=IN_M(i);
for each symbolic register operand $s_k$ of instruction i (Suppose $s_k$ is the Nth operand)
  Find out the value of PrevAssign where ($s_k$ PrevAssign) $\in$ OUT_M(i);
  CurAssign = Regclass ($s_k$, i);
  if (CurAssign = =C)
        if (PrevAssign! =C)
            if (IsValidRegClassAssignment (i, Nth, PrevAssign))
                Regclass($s_k$ i)=PrevAssign;
                continue;       /*continue the next loop iteration */
            else
                CurAssign=GetNextRegClass(Inst, NthOperand);
                If ($s_k$ is not the destination operand)
                    Insert before i the register class fixup from PrevAssign to CurAssign;
        else
            CurAssign =GetNextRegClass(Inst, NthOperand);
        Regclass ($s_k$, i) =CurAssign;
        Replace ($s_k$ PrevAssign) with ($s_k$ CurAssign) in OUT_M(i);
   else
        if  (($s_k$, CurAssign) $\notin$ OUT_M(i))
                if (PrevAssign!=C AND $s_k$ is not the destination operand)
                     insert before i the register class fixup from PrevAssign to CurAssign;
                Replace ($s_k$ PrevAssign) with ($s_k$ CurAssign) in OUT_M(i);
```

Fig. 7